Wireless Internet: TCP in Wireless

Module W.int.2

Dr.M.Y.Wu@CSE
Shanghai Jiaotong University
Shanghai, China

Dr.W.Shu@ECE
University of New Mexico
Albuquerque, NM, USA

1

W.int.2-2

Wireless TCP

- Internet + wireless + mobility
- Mobile IP: network layer, module W.int.1
- *Wireless TCP: transport layer*
  - *TCP mechanism*
  - TCP mechanism with wireless & mobility
  - Split connection & Indirect TCP
  - Snoop TCP
- End of module W.int.2
- WAP: application layer, module W.int.3

W.int.2-3

TCP mechanism

- TCP is designed for
  - Fixed end-systems
  - Stationary, wired networks
- TCP congestion control in non-wireless networks
  - Packet loss typically due to (temporary) overload situations
  - Router has to discard packets when the buffers are full
  - TCP detects congestion only indirect via missing ACKs or duplicated ACKs.
- However, such presumption can be totally non true with wireless + mobile
  - Packet loss due to commonly high transmission errors
  - Packet loss due to change of network

W.int.2-4

TCP mechanism

- Review: TCP slow-start algorithm
  - Start with a congestion window size equal to one segment
  - Exponential increase of the congestion window up to the congestion threshold, then linear increase
  - Missing ACK causes the reduction of the congestion threshold to $\frac{1}{2}$ of the current congestion window
  - Congestion window starts again with one segment
- Review: TCP fast retransmit/fast recovery
  - TCP sends an ACK only after receiving a packet
  - If a sender receives duplicated ACKs for the same packet, this is due to a gap in received packets at the receiver
  - Duplicated ACKs is not as bad as missing ACKs
  - Therefore, packet loss is not due to congestion, continue with current congestion window (do not use slow-start)

W.int.2-5

Wireless TCP

- Internet + wireless + mobility
- Mobile IP: network layer, module W.int.1
- Wireless TCP: transport layer
  - TCP mechanism
  - *TCP mechanism with wireless & mobility*
  - Split connection & Indirect TCP
  - Snoop TCP
- End of module W.int.2
- WAP: application layer, module W.int.3

W.int.2-6

TCP mechanisms: wireless & mobility

- Impacts on TCP by introducing wireless & mobility
  - wired networks:
    - packet loss due to congestions
    - Packet reordering, but not often
  - wireless networks:
    - bit errors due to wireless channel impairments
    - handoffs due to mobility
      - e.g. a MN roams from one access point to another while there are still packets in transit to the wrong access point
    - possibly congestion, but not often

W.int.2-7

TCP mechanisms: wireless & mobility

- Impacts on TCP by introducing wireless & mobility
  - Bursts of errors may occur due to low signal strength or longer period of noise
    - More than one packet lost in TCP
    - More likely to be detected as a timeout ⇒ enter slow start!
  - Delay is often very high
    - Round-trip time can be very long and variable
    - TCP's timeout mechanisms may not work well
    - Problem exacerbated by link-level retransmission
  - Links may be asymmetric
    - Delayed ACKs in the slow direction can limit throughput in the fast direction

W.int.2-8

Wireless TCP mechanism

- TCP is a complex protocol
  - Minimal support from underlying protocols
  - Indirect observation of network environment
  - Large number of competing flows from different hosts
  - Congestion avoidance is still a research issue
- TCP with wireless?
  - TCP does not perform well in a wireless environment where packets are usually lost due to bit errors, not congestion
  - However, TCP cannot be easily changed/modified to adapt to wireless & mobility!

W.int.2-9

Wireless TCP mechanism

- Can we solve it at link layer? hiding wireless loss from TCP
  - FEC can be used to correct small number of errors
  - Error detection, then retransmission at link layer
    - How to detect?
      - Link layer timeout, or ACKs, NACKs.
- Can we solve it at transport layer?
  - Split connection
  - TCP-aware link layer retransmission

W.int.2-10

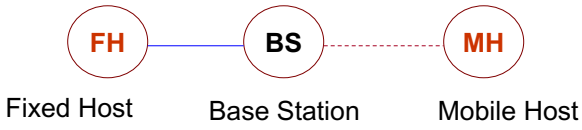
Wireless TCP

- Internet + wireless + mobility
- Mobile IP: network layer, module W.int.1
- Wireless TCP: transport layer
  - TCP mechanism
  - TCP mechanism with wireless & mobility
  - *Split connection & Indirect TCP*
  - Snoop TCP
- End of module W.int.2
- WAP: application layer, module W.int.3

W.int.2-11

Wireless TCP: split connection

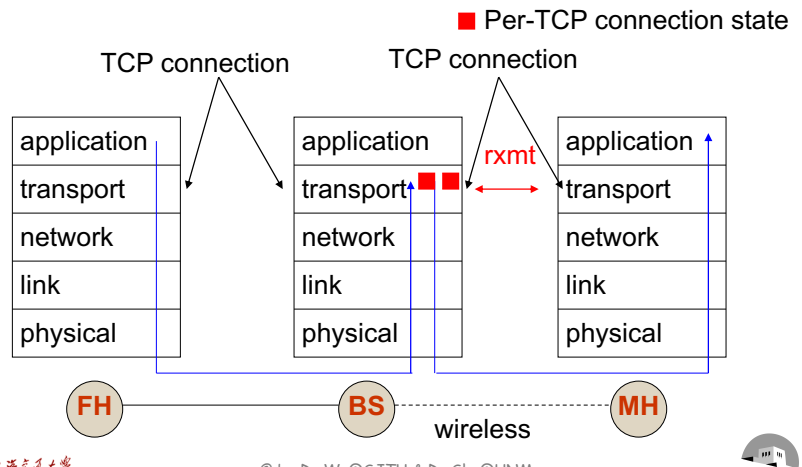
- Split connection
  - A single TCP connection split into one connection on the wired part of route and one over wireless part of the route
    - If wireless link is not last on route, then more than two TCP connections may be needed
  - Connection between wireless host MH and fixed host FH goes through base station BS
    - FH-MH = FH-BS + BS-MH



W.int.2-12

Wireless TCP: split connection

- Split connection protocol structure



W.int.2-13

Wireless TCP: split connection

- Split connection
  - Split connection results in independent flow control for the two parts
    - Flow/error control protocols, packet size, time-outs, may be different for each part
  - FH-MH = FH-BS + BS-MH, primary responsibility at base station
    - FH-BS: regular TCP
    - BS-MH: a protocol optimized for wireless link
      - Indirect TCP (I-TCP), Bakre 1995
      - Asymmetric transport protocol (Mobile-TCP), Haas 1997
      - Selective repeat protocol (SRP) on top of UDP, Yavatkar 1994

W.int.2-14

Wireless TCP: split connection

- Typical work:

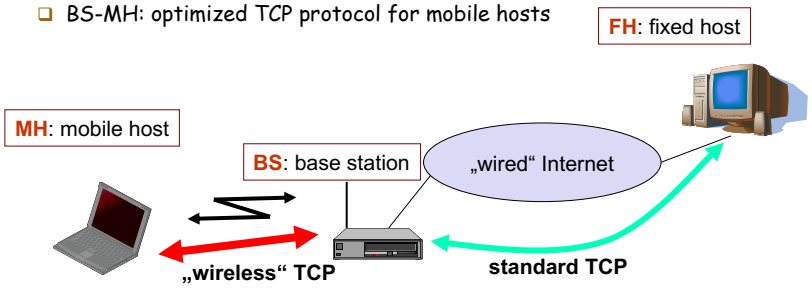
Indirect TCP

- A. Bakre and B. Badrinath, I-TCP: Indirect TCP for Mobile Hosts, 15th Int'l Conf. on Distributed Computing Systems (ICDCS), 1995.
- A. Bakre and B. Badrinath, Implementation and performance evaluation of Indirect TCP, IEEE Trans. on Computers, 46(3), March 1997.

W.int.2-15

Wireless TCP: split connection

- Indirect TCP
  - No real end-to-end connection any longer: FH – BS – MH
  - FH-BS: regular TCP
    - hosts in the fixed part of the network do not notice the characteristics of the wireless part
  - BS-MH: optimized TCP protocol for mobile hosts



W.int.2-16

Wireless TCP: split connection

- Advantages
  - FH-BS: No changes in the fixed network necessary
    - All current optimizations to TCP still work
  - BS-MH: connection can be optimized independently
    - Different flow/error control can be applied
    - Selective ACKs improve performance for such cases
  - Local recovery of errors
    - Transmission errors on the wireless link do not propagate into the fixed network
    - Faster recovery due to relatively shorter RTT on wireless link

W.int.2-17

Wireless TCP: split connection

- Negative impacts
  - End-to-end semantics violated
    - ACK may be delivered to sender, before data delivered to the receiver
  - BS retains hard state
    - BS failure can result in loss of data (unreliability)
    - Hand-off latency increases due to state transfer
  - Buffer space needed at BS for each TCP connection
    - BS buffers tend to get full, when wireless link slower
    - Window on BS-MH connection reduced in response to errors
  - Extra copying of data at BS
    - increases end-to-end latency

W.int.2-18

Wireless TCP: split connection

- Indirect TCP's alternative: Mobile TCP
  - Special handling of lengthy and/or frequent disconnections
    - If I-TCP, data buffered grows too large in size
    - How to improve? Informing the sender!
  - M-TCP splits as I-TCP does
    - unmodified TCP fixed network to supervisory host (SH)
    - optimized TCP SH to MH
  - Supervisory host
    - no caching, no retransmission
    - monitors all packets, if disconnection detected
      - set sender window size to 0 to advertise back to the sender
    - Resume it when MH can be connected again
  - Good: maintains semantics, supports disconnection, no buffering
  - Bad: loss on wireless link propagated into fixed network

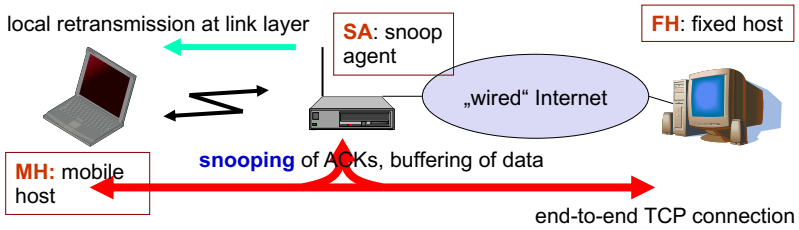
W.int.2-19

Wireless TCP

- Internet + wireless + mobility
- Mobile IP: network layer, module W.int.1
- Wireless TCP: transport layer
  - TCP mechanism
  - TCP mechanism with wireless & mobility
  - Split connection & Indirect TCP
  - *Snoop TCP*
- End of module W.int.2
- WAP: application layer, module W.int.3

W.int.2-20

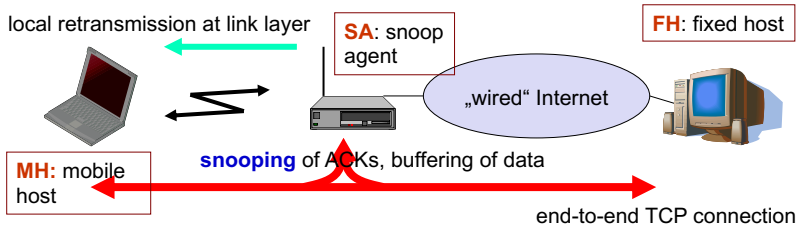
Wireless TCP: snoop TCP



- Snoop TCP: Hari Balakrishnan, Srinivasan Seshan, Randy H. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks ACM Wireless Networks, 1(4), December 1995.
- Basic idea: soft-state in BS's (the snoop agent) is used to perform local retransmissions across a wireless link to avoid end-to-end congestion control and retransmissions
- Software: The Berkeley Snoop Protocol

W.int.2-21

Wireless TCP: snoop TCP

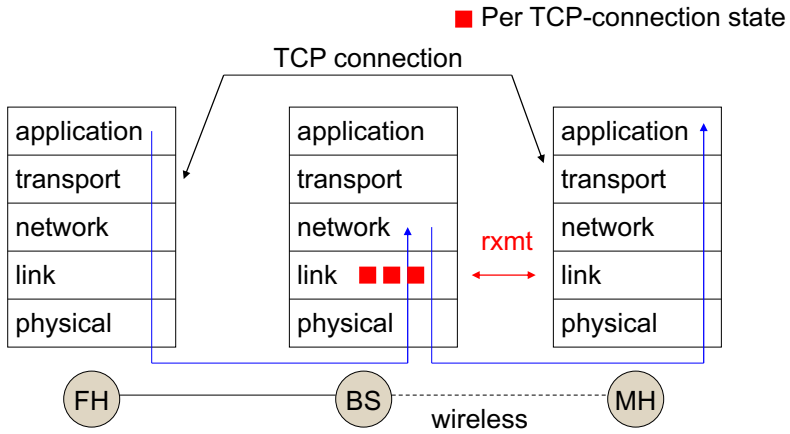


- Transparent extension of TCP within the snoop agent (SA)
  - changes of TCP only within the SA, packets are buffered
  - Local retransmission: lost packets on the wireless link will be retransmitted immediately by the MH or SA, respectively
  - ACK snooping/filtering: SA "snoops" the packet flow and recognizes ACKs in both directions, it also filters duplicated ACKs

W.int.2-22

Wireless TCP: snoop TCP

- Snoop TCP protocol structure



W.int.2-23

Wireless TCP: snoop TCP

- Data transfer FH → (SA) → MH
  - SA buffers data until it receives ACK of the MH
  - SA detects packet loss via
    - duplicated ACKs
      - SA discards duplicate ACKs from mobile host
    - time-out (can be relatively short)
  - Fast retransmission possible at link layer, transparent for the fixed network
  - SA, however, cannot generate ACKs that are sent back to FH
    - Unlike I-TCP, Snoop TCP preserves end-to-end TCP semantics

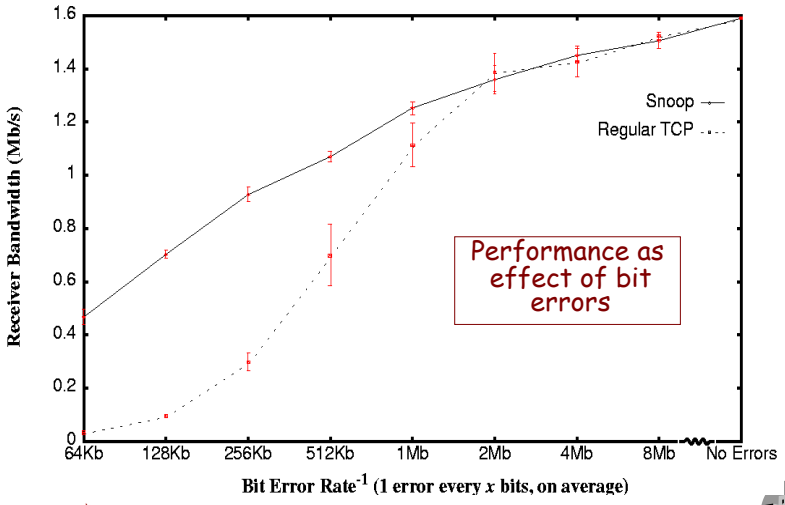
W.int.2-24

Wireless TCP: snoop TCP

- Data transfer MH → (SA) → FH
  - SA detects packet loss on the wireless link via sequence numbers, SA answers directly with a NACK to the MH
  - MH can now retransmit data with only a very short delay, hopefully in time to avoid a TCP timeout at FH
- Summary: provide reliable link layer that is TCP aware
  - SA at the access point or foreign agent
  - Buffers data at the ends of the links for retransmissions
  - Snooping on ACKs and filters duplicate ACKs

W.int.2-25

Wireless TCP: snoop TCP



W.int.2-26

Wireless TCP: snoop TCP

- Advantages
  - High throughput can be achieved
  - Local recovery from wireless losses
  - Fast retransmit not triggered at sender despite out-of-order link layer delivery
  - Preserves end-to-end TCP semantics
  - Soft state at base station
    - loss of the soft state affects performance, but not correctness
    - Automatic fallback to standard TCP
  - No need to ensure that all foreign networks provide a Snoop agent

W.int.2-27

Wireless TCP: snoop TCP

- Negative impacts
  - Does not isolate the wireless link as good as I-TCP
  - Violate the general layering approach
    - Link layer at base station needs to be TCP-aware
  - Snooping might be useless depending on encryption schemes
    - E.g., not useful if TCP headers are encrypted (IPsec)
  - The TCP stack on the MH had to be modified to handle NACKs for reverse (MH→FH) traffic
  - Cannot be used if TCP data and TCP ACKs traverse different paths (both do not go through the base station)

W.int.2-28

Wireless TCP

- Internet + wireless + mobility
- Mobile IP: network layer, module W.int.1
- Wireless TCP: transport layer
  - TCP mechanism
  - TCP mechanism with wireless & mobility
  - Split connection & Indirect TCP
  - Snoop TCP
- *End of module W.int.2*
- WAP: application layer, module W.int.3